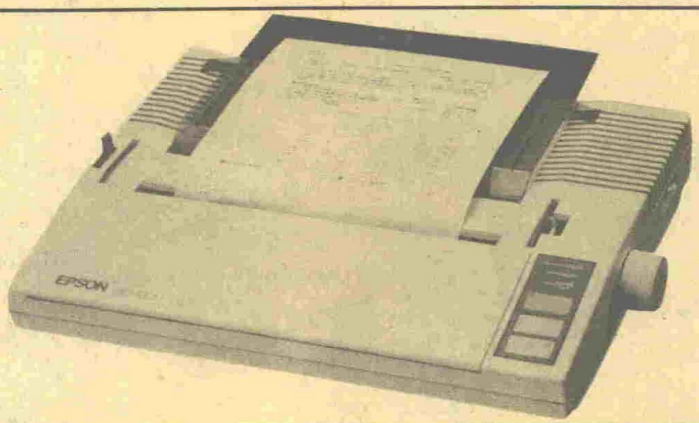
แก้ปัญหา

การพิมพ์กรอบ



อนุชาติ ลาสนธิ

การแสดงผลไม่ว่าจะเป็นทางจอภาพ (Screen) หรือเครื่องพิมพ์ (Printer) บางครั้งก็มีความต้องการที่จะจัดแบ่งข้อมูลที่ให้แสดงผลออกมาให้ดูง่ายขึ้น โดยใช้กรอบหรือตารางเป็นตัวแบ่งข้อมูลเหล่านั้น (ต่อไปผู้เขียนจะใช้คำว่า "กรอบ" ก็ให้เข้าใจว่า หมายถึง ตารางด้วย)

ชื่อเรื่องก็บอกแล้วว่าเป็นการแก้ปัญหาเรื่องการพิมพ์กรอบ ความหมายของ "การแก้ปัญหา" ของผู้เขียนก็คือ

1. สามารถรับตัวอักษร (Character) ที่ใช้ในการพิมพ์กรอบ เข้ามาทางคีย์บอร์ดได้โดยตรง

2. กรอบที่พิมพ์ออกทางเครื่องพิมพ์ ก็สามารถพิมพ์เป็นกรอบที่สมบูรณ์

เนื่องจากบทความของผู้เขียนมีสองเรื่อง ถ้าจะพูดเคล้ากันไป ผู้อ่านอาจจะไม่เข้าใจเท่าที่ควร จึงขอกล่าวทีละเรื่อง เรื่องแรกจะกล่าวถึงการแก้ปัญหาทางเครื่องพิมพ์ และตามด้วยคีย์บอร์ด, จอภาพ

ตัวอักษรที่ใช้พิมพ์กรอบของเครื่องพิมพ์

การพิมพ์กรอบออกทางจอภาพ ก็คงจะไม่มีปัญหาอะไร ถ้าต้องการออกเครื่องพิมพ์ มีปัญหาแน่ครับ

1. ถ้าเครื่องพิมพ์นั้นมีรหัสภาษาไทยด้วย กรอบที่พิมพ์ออกมาจะกลายเป็นภาษาไทย

2. กรณีที่เครื่องพิมพ์นั้นไม่มีรหัสภาษาไทย ก็จะพิมพ์กรอบได้แต่กรอบด้านแนวตั้งจะไม่ติดกัน

ปกติแล้วรอมคาแรกเตอร์ (รอมที่ใช้เก็บตัวอักษร) ของเครื่องพิมพ์ จะต้องมีตัวอักษรที่ใช้ในการพิมพ์กรอบด้วย แต่ก็คงจะเหมือนคำพังเพยว่า "เข้าเมืองตาหลิ่วต้องหลิ่วตาตาม" คือเครื่องพิมพ์ที่อยู่ในเมืองไทยก็ควรจะพิมพ์ภาษาไทยได้ด้วย (ถึงตรงนี้ผู้อ่านที่มีเครื่องพิมพ์ ที่ไม่มีรหัสภาษาไทยคงจะคิดว่าเป็นการแก้ปัญหาเฉพาะเครื่องพิมพ์ที่มีภาษาไทยเท่านั้น ไม่ใช่อย่างที่คุณคิดครับ ลองอ่านต่ออีกนิด แล้วจะได้โปรแกรม

SOFTMAIL

ดี ๆ ไว้ใช้งานกัน) ดังนั้นจึงมีการแก้ไขตัวอักษรเดิมจากค่าแอสกี้ 128-255 ให้กลายเป็นภาษาไทย ซึ่งตัวอักษรที่ใช้ในการพิมพ์กรอบก็รวมอยู่ในนี้ด้วย ผลที่ตามมาก็คือ ตัวอักษรที่มีค่ารหัสแอสกี้ตั้งแต่ 128-255 เมื่อส่งออกไปพิมพ์ที่เครื่องพิมพ์ก็จะถูกพิมพ์เป็นภาษาไทย เมื่อเป็นเช่นนี้การแก้ปัญหาการพิมพ์กรอบ ก็คงต้องใช้หลักการสร้างตัวอักษรใหม่ขึ้นเอง (โดยไม่ต้องเข้าไปแก้ที่รอมคาแรกเตอร์ ของเครื่องพิมพ์) หรือเรียกกันอีกอย่างหนึ่งว่าการ DOWNLOAD นั่นเอง แต่ก็มีปัญหาตามมาอีกว่า เครื่องพิมพ์บางรุ่นไม่สามารถทำ DOWNLOAD

ยังไม่หมดหนทางแค่นี้ เนื่องจากว่าคนที่ออกแบบภาษาไทยให้กับเครื่องพิมพ์ ก็มีหลายรหัส เช่นรหัสของ มหาวิทยาลัยเกษตรฯ, สมอ. ซึ่งแต่ละรหัสก็แตกต่างกันออกไป แล้วแต่ความคิดของใคร แต่ที่เหมือนกันคือ ยังคงเหลือตัวอักษรที่ใช้ในการพิมพ์กรอบไว้ให้ 1 ชุด และค่ารหัสแอสกี้ เปลี่ยนไปตามรหัสภาษาไทยนั้น ๆ ดูตารางที่ 1.

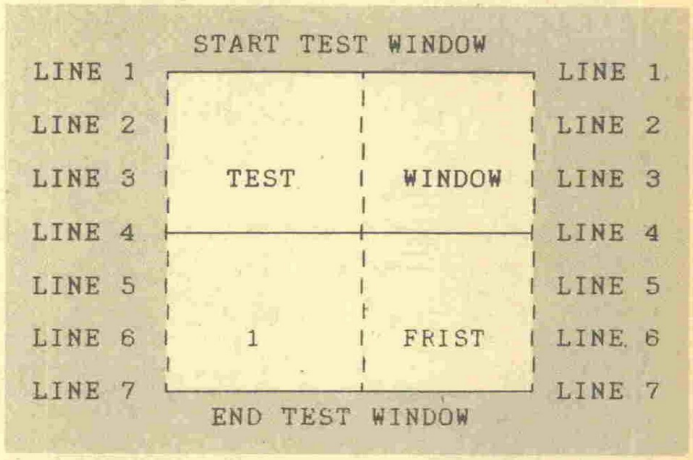
	เกษตร		สมอ.	
CHAR	DEC	HEX	DEC	HEX
│	132	84	149	95
─	133	85	150	96
┌	128	80	152	98
┬	137	89	157	9D
┐	129	81	156	9C
├	134	86	151	97
┼	138	8A	158	9E
┤	135	87	154	9A
└	130	82	159	9F
┴	136	88	155	8B
┘	131	83	153	99

ตารางที่ 1 ตัวอักษรที่ใช้พิมพ์กรอบของเครื่องพิมพ์ รหัสเกษตร, สมอ.

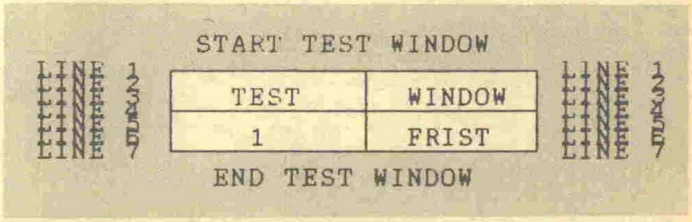
หลักการ

ความคิดในการแก้ปัญหาเริ่มแรกของผู้เขียนคือ ต้องเขียนโปรแกรมฝังตัวในหน่วยความจำ (Resident program) โปรแกรมนี้จะทำหน้าที่คอยตรวจสอบตัวอักษรที่ส่งไปให้เครื่องพิมพ์ ว่าเป็นตัวอักษรที่ใช้พิมพ์กรอบหรือไม่ ถ้าไม่ใช่ก็จะส่งให้เครื่องพิมพ์โดยไม่เปลี่ยนแปลงค่าใด ๆ แต่ถ้าใช่ จะต้องเปลี่ยนค่าแอสกี้ตัวนี้ให้เป็นตัวใหม่ที่ตรงกับตัวอักษรที่ใช้พิมพ์กรอบของเครื่องพิมพ์ แล้วจึงส่งไปให้เครื่องพิมพ์ เช่น ถ้าโปรแกรมส่งค่า '|' ซึ่งมีค่าแอสกี้ = (179) โปรแกรมนี้จะต้องเปลี่ยนค่าแอสกี้เป็น 132 (เมื่อใช้รหัสเกษตร) หรือ 149 (เมื่อใช้รหัส สมอ.) เครื่องพิมพ์ก็จะสามารถพิมพ์ '|' ได้ และตัวอื่น ๆ ก็เช่นกัน

เมื่อเขียนโปรแกรมเสร็จ (ตั้งชื่อโปรแกรมว่า PRNDRV.COM) และทดลองรันดู พร้อมกับได้ทดลองส่งกรอบไปให้เครื่องพิมพ์ พิมพ์ได้ผลครับ คือเครื่องพิมพ์สามารถพิมพ์กรอบออกมาได้ แต่กรอบทางแนวตั้ง (vertical) ไม่ติดกัน ดังรูปที่ 1 เนื่องมาจากปกติเครื่องพิมพ์จะถูกเซตไว้ให้พิมพ์แต่ละบรรทัดห่างกัน หรือค่า Line spacing นั่นเอง ปัญหานี้แก้ได้โดยเซตค่า Line spacing ใหม่ ให้พิมพ์แต่ละบรรทัดติดกัน เมื่อทดลองรันดูอีกครั้ง สามารถพิมพ์กรอบติดกันได้ แต่ถ้ากรณีที่มีตัวอักษรอยู่ข้างกรอบ จะทำให้ตัวอักษรแต่ละบรรทัดติดกันด้วย ดังรูปที่ 2



รูปที่ 1 กรอบด้านแนวตั้งไม่ติดกัน



รูปที่ 2 กรอบแนวตั้งติดกันทำให้ตัวอักษรแต่ละบรรทัดติดกันด้วย

การทำให้เครื่องพิมพ์สามารถพิมพ์กรอบได้สมบูรณ์นั้น ถ้าหากผู้อื่นทำอาจจะใช้วิธีอื่นที่แตกต่างกันไป แต่ผู้เขียนใช้หลักการนี้ครับ

1. ตัวอักษรทุกตัวที่จะส่งไปให้เครื่องพิมพ์ จะเก็บไว้ในหน่วยความจำส่วนหนึ่งก่อน ตั้งชื่อว่า BUFFER1 จนกว่าจะได้รับรหัส RETURN (13)

2. เมื่อได้รับรหัส RETURN แล้วจะทำการตรวจสอบค่าใน BUFFER1 ว่ามีตัวอักษรที่ใช้ในการพิมพ์กรอบหรือไม่ ถ้าไม่มีจะเซต line spacing ให้เป็นปกติ แล้วส่งค่าใน BUFFER1 ไปพิมพ์ ถ้ามีจะตรวจสอบอีกว่ามี '|', '‖' หรือไม่ ถ้าไม่มี ก็จะเปลี่ยนตัวอักษรที่ใช้พิมพ์กรอบให้ตรงกับเครื่องพิมพ์ แล้วส่งค่าใน BUFFER1 ไปพิมพ์ แต่ถ้ามี '|', '‖' ก็จะเก็บรหัส RETURN (13), LINE FEED (10) และ '|', '‖' ไว้ที่ตำแหน่งเดียวกันใน BUFFER2 พร้อมทั้งเปลี่ยนค่าให้ตรงกับรหัสของเครื่องพิมพ์ เซต line spacing ให้พิมพ์ติดกัน ส่งค่าใน BUFFER2 ไปพิมพ์ ตามด้วย BUFFER1

3. เมื่อออกจากโปรแกรมเซต line spacing ให้พิมพ์ปกติ ตัวอย่างเช่น

- writeln (LST, 'ABCDEF'); (ภาษา Pascal) เป็นการส่ง 'ACDEF' ไปยังเครื่องพิมพ์ ในกรณีนี้ไม่มีตัวอักษรที่ใช้พิมพ์กรอบ 'ABCDEF' ก็จะถูกส่งไปให้เครื่องพิมพ์ตามค่าเดิม ไม่มีการเปลี่ยนแปลงค่าใด ๆ

- writeln (LST, 'xyz ┌─┐ mnop'); กรณีนี้โปรแกรม PRNDRV จะเปลี่ยนเฉพาะตัวอักษรที่ที่พิมพ์กรอบคือ '┌─┐' ให้ตรงกับรหัสพิมพ์กรอบของเครื่องพิมพ์เท่านั้น แล้วส่งไปพิมพ์

- writeln (LST, 'AB yz'); กรณีเช่นนี้ มีตัวอักษรที่ใช้พิมพ์กรอบ '|' อยู่ด้วย โปรแกรม PRNDRV จะเก็บค่าใน BUFFER2 เมื่อเทียบกับค่าใน BUFFER1 ดังรูปที่ 3

แล้วจะทำการเปลี่ยนค่าตัวอักษรที่ใช้พิมพ์กรอบใน BUFFER1, BUFFER2 ให้ตรงกับรหัสที่เครื่องพิมพ์จะเปลี่ยนดังรูปที่ 4

การพิมพ์ครั้งแรกจะพิมพ์ค่าใน BUFFER2 ก่อน แล้วจึงส่ง BUFFER1 ออกไปพิมพ์ หลักการทั้งหมดของผู้เขียนก็มีแค่นี้ ส่วนโพรซีเดอร์ GETKEY เป็นโพรซีเดอร์เกี่ยวกับการเลือกและแสดงเมนูเท่านั้น

อธิบายโปรแกรม

PRNDRV PROC FAR :

- row, column เก็บตำแหน่งของเคอร์เซอร์ที่จะกระโดดไป

- position, shape เก็บตำแหน่ง และขนาดเคอร์เซอร์เดิม ก่อนการเรียกเมนู

- oldvect17H, oldvect16H เก็บ interrupt vector เก่าของ INT 17H, INT 16H ไว้

- seleccode เก็บค่าตัวเลือกเมนู ON/OFF ปกติเป็น ON (78 = 'N')

- codeprn เก็บการเลือกรหัสของเครื่องพิมพ์ ปกติเป็น สมอ. (83= 'S')

- numprn เครื่องพิมพ์ตัวที่กำลังติดต่อ

- countbuf จำนวนตัวอักษรที่ส่งไปพิมพ์แต่ละครั้ง

- statusact เก็บสถานะของโปรแกรมว่าเป็น ON หรือ

```
BUFFER1 = 65,66,218,196,194,191,179,121,122,10,13
            A B  ┌  ─  ┬  ┐  │  y  z  LF CR
BUFFER2 = 32,32, 32, 32, 32, 32,179, 32, 32,10,13
          . . . . . . . . . . . │  . . . 'LF CR
```

รูปที่ 3

```
┌ = 218 เปลี่ยนเป็น 128 (เกษตร)  หรือ  เปลี่ยนเป็น 152 (สมอ.)
─ = 196 เปลี่ยนเป็น 133 (เกษตร)  หรือ  เปลี่ยนเป็น 150 (สมอ.)
┬ = 194 เปลี่ยนเป็น 137 (เกษตร)  หรือ  เปลี่ยนเป็น 157 (สมอ.)
┐ = 191 เปลี่ยนเป็น 129 (เกษตร)  หรือ  เปลี่ยนเป็น 153 (สมอ.)
│ = 179 เปลี่ยนเป็น 132 (เกษตร)  หรือ  เปลี่ยนเป็น 149 (สมอ.)
```

รูปที่ 4

OFF ปกติจะเป็น ON มีค่า 0FFH

- statusbuf บอกสถานะว่าตอนนี้ ได้รับรหัส RETURN หรือยัง

- statcheck บอกสถานะว่าตัวอักษรที่ส่งให้เครื่องพิมพ์มีตัวอักษรที่ใช้พิมพ์กรอบหรือไม่ ถ้ามี = 0FFH

- versmo, horsmo เป็นค่าคงที่ เก็บตัวอักษร vertical, horizontal bar ของ สมอ.

- verku, horku เป็นค่าคงที่ เก็บตัวอักษร vertical, horizontal bar ของ เกษตร

- markker ใช้ตรวจสอบว่าโปรแกรม PRNDRV.COM นี้ RUN แล้วหรือยัง เพื่อป้องกัน RUN ซ้ำ

- คำสั่ง or ah, ah เป็นการตรวจสอบว่าเป็น INT 17H Function 0 หรือไม่

หน้าที่ต่าง ๆ ของโปรแกรมย่อยดูในตารางที่ 2

GETKEY PROC FAR :

- or ah, ah ตรวจสอบว่าเป็น INT 16H function 0 หรือไม่

- cmp ah, 39 คือคีย์ Shift + F10 กดคีย์นี้เพื่อเรียกเมนูขึ้นมาเลือกสถานะต่าง ๆ

หน้าที่ต่าง ๆ ของโปรแกรมย่อยดูในตารางที่ 3

โปรแกรมย่อย	หน้าที่
SAVE_STRING	เก็บตัวอักษรที่จะส่งไปเครื่องพิมพ์ไว้ใน BUFFER1 จนกว่าจะได้รับ RETURN
SENDNORMAL	ส่ง Control code ให้พิมพ์ไปกลับ (bi-direction), set line spacing เป็น 1/6″ หรือให้พิมพ์แต่ละบรรทัด ห่างกันปกติ
SEND_672	ส่ง Control code ให้พิมพ์ทางเดียว (uni-direction), พิมพ์จากซ้ายไปขวา (left to right), set line spacing เป็น 6/72″ หรือให้พิมพ์แต่ละบรรทัดติดกัน
PRINTEBAR	พิมพ์เฉพาะตัวอักษรที่ใช้พิมพ์กรอบทางแนวตั้ง (vertical bar) ที่เก็บใน BUFFER2
PRINTBUFFER	พิมพ์ตัวอักษรปกติและตัวอักษรที่ใช้พิมพ์กรอบ ที่เก็บใน BUFFER1
SAVEBAR	ค้นหา กรอบแนวตั้ง (vertical, double vertical bar), LINE FEED, RETURN และเก็บไว้ใน BUFFER2
FIND_GEN	เปลี่ยนรหัสที่จะพิมพ์ให้ตรงกับรหัสของเครื่องพิมพ์
TOPRN	ส่งตัวอักษร 1 ตัวไปยังเครื่องพิมพ์
FIND_BAR	ตรวจสอบว่าตัวอักษรที่รับเข้ามาเป็นตัวอักษรที่ใช้พิมพ์กรอบหรือไม่
CHECK_VERBAR	ตรวจสอบค่าที่รับมาว่าเป็น vertical, double vertical bar ถ้าใช่ก็จะเปลี่ยนให้ตรงกับรหัสเครื่องพิมพ์ด้วย
CHECK_HORBAR	ตรวจสอบค่าที่รับมาว่าเป็น horizontal, double horizontal bar ถ้าใช่ก็เปลี่ยนให้ตรงกับรหัสเครื่องพิมพ์ด้วย

ตารางที่ 2 หน้าที่ต่าง ๆ ของโปรแกรมย่อย PRNDRV

โปรแกรมย่อย	หน้าที่
SAVESCREEN	เก็บจอภาพที่กำลังแสดงผลนั้นไว้ ก่อนที่จะแสดงเมนู
RESTSCREEN	คืนจอภาพเดิมให้ หลังออกจากเมนู
SAVE_CURSOR	เก็บตำแหน่งและขนาดของ cursor ขณะนั้นไว้ ก่อนที่จะแสดงเมนู
SET_CURSOR	ให้ cursor กระโดดไปตำแหน่งที่เก็บค่าไว้ใน row, column
WRITEASK	พิมพ์ข้อความที่กำหนดไว้ชื่อว่า ASK หรือข้อความที่แสดงเมนูนั่นเอง ออกทางจอภาพ
WRITESELEC	นำค่าตัวเลือกที่กำหนดไว้เริ่มแรกมาแสดง
SELEC_ONOFF	เลือกสถานะของโปรแกรม PRNDRV.COM ว่าจะให้เป็น ON หรือ OFF
SELEC_CODE	เลือกรหัส KU, SMO, NOTHING
ONE_CHAR	หยุดรอรับคีย์ 1 ตัวอักษร
RES_CURSOR	คืนตำแหน่ง และขนาดของ CURSOR หลังออกจากเมนู
UPERCASE	เปลี่ยนตัวอักษรตัวเล็กให้เป็นตัวใหญ่

ตารางที่ 3 หน้าที่ต่าง ๆ ของโปรแกรมย่อย GETKEY

```
..... PRNDRV:COM      o(N)  or  of(F) ..... [N/F]?
Printer code ----------------------------------------
                (K)U,(S)MO,(N)othing ..... [K/S/N]?
```

รูปที่ 5 เมนูของโปรแกรม PRNDRV.COM

ค่าที่กำหนดไว้เป็นตาราง

- window ตัวอักษรที่ใช้พิมพ์กรอบทั้งหมดของจอภาพ ยกเว้น vertical, double vertical
- smo ตัวอักษรที่ใช้พิมพ์กรอบของเครื่องพิมพ์รหัส สมอ.
- ku ตัวอักษรที่ใช้พิมพ์กรอบของเครื่องพิมพ์รหัสเกษตร
- bartable ตัวอักษรที่ใช้พิมพ์กรอบ เพื่อช่วยวิเคราะห์การพิมพ์กรอบ
- fourwindow ตัวอักษรที่ใช้พิมพ์กรอบ กำหนดไว้เพื่อช่วยวิเคราะห์การพิมพ์กรอบทั้ง 4 รูปแบบ
- bi__direction เป็น control code ของเครื่องพิมพ์ให้พิมพ์ไปกลับ
- uni__direction เป็น control code ของเครื่องพิมพ์ให้พิมพ์ไปทางเดียว
- left__right เป็น control code ของเครื่องพิมพ์ ให้พิมพ์จากซ้ายไปขวา
- inch__16 เป็น control code ของเครื่องพิมพ์ ให้พิมพ์แต่ละบรรทัดห่างกัน 1/6 นิ้ว
- inch__672 เป็น control code ของเครื่องพิมพ์ ให้พิมพ์แต่ละบรรทัดห่างกัน 6/72 นิ้ว
- buffer 1, buffer 2 กำหนดไว้เพื่อเก็บตัวอักษรที่จะส่งให้เครื่องพิมพ์
- ask กำหนด string เพื่อแสดงเมนู
- โปรแกรมที่อยู่หลัง INSTALL ทำให้ PRNDRV.COM ฝังในหน่วยความจำ (resident)

การใช้งาน

1. เมื่อรันโปรแกรม PRNDRV.COM จอภาพที่แสดงผลขณะนั้นจะถูกลบ และมีข้อความแสดงที่มุมบนด้านซ้ายของจอภาพ เป็นการแนะนำตัวเองนิดหน่อย

2. กด Shift + F10 จะมีเมนูขึ้นมาให้เลือก หน้าตาของเมนูดูรูปที่ 5 ตัวเลือกตัวแรกจะเป็นการถามเพื่อตั้งสถานะของโปรแกรม PRNDRV.COM เองว่าจะให้เป็น ON หรือ OFF (ปกติเป็น 'N' = ON) ถ้าเลือก OFF กด 'F' หรือ 'f' แสดงว่าไม่ต้องการให้โปรแกรม PRNDRV.COM ทำงาน แต่เครื่องพิมพ์จะทำงานปกติทุกอย่าง เพียงแต่การพิมพ์ไม่ขึ้นกับโปรแกรม PRNDRV.COM ถ้าส่งกรอบออกพิมพ์จะไม่สามารถพิมพ์กรอบได้ ดังรูปที่ 6 ถ้าเลือก 'ON' กด 'N' หรือ 'n' ค่าที่จะส่งไปพิมพ์จะต้องผ่านการวิเคราะห์ เพื่อหาตัวอักษรที่พิมพ์กรอบ โดยโปรแกรม PRNDRV เสียก่อน เมื่อส่งกรอบออกไปพิมพ์ที่เครื่องพิมพ์ ก็จะพิมพ์กรอบออกได้โดยสมบูรณ์ แต่การพิมพ์จะช้าลง เนื่องจากการพิมพ์กรอบถ้าเซตให้เครื่องพิมพ์ พิมพ์แบบไปกลับ (bi-direction) จะทำให้พิมพ์กรอบไม่ตรง ดังรูปที่ 7 ต้องเซตให้เครื่องพิมพ์ พิมพ์ไปด้านเดียว (uni-direction) จึงจะพิมพ์กรอบได้ตรงดังรูปที่ 8. ในการเลือกเมนูถ้าไม่ต้องการเปลี่ยนตัวเลือก กด Esc ผ่านไป

```
          START TEST WINDOW
LINE 1 ·ฤฤฤ
```

3. เลือกรหัสเครื่องพิมพ์มีให้เลือกได้ 3 ตัวเลือก (K)U รหัส เกษตร, (S)MO รหัส สมอ., (N) othing เมื่อเครื่องพิมพ์นั้นไม่มีรหัสภาษาไทย

เนื่องจากตัวอักษรที่ใช้พิมพ์กรอบมี 4 รูปแบบ ดังรูปที่ 9. ดังนั้น ถ้าเครื่องพิมพ์ที่ไม่มีรหัสภาษาไทย จะสามารถพิมพ์ออกเครื่องพิมพ์ได้ทั้ง 4 รูปแบบเหมือนที่จอภาพแสดงได้ แต่ถ้าเครื่องพิมพ์ที่มีรหัสเป็นภาษาไทย ตัวอักษรที่ใช้พิมพ์กรอบเหลือเพียงชุดเดียว ไม่ว่าจะส่งกรอบรูปแบบไหนไปพิมพ์ เครื่องพิมพ์ก็จะพิมพ์ให้ได้รูปแบบเดียว คือรูปแบบแรก

โปรแกรม PRNDRV.COM เป็นโปรแกรมฝังตัวในหน่วยความจำ (resident program) จะทำงานเมื่อเกิด INT 17H function 0 เท่านั้น (เมื่อมีการส่งตัวอักษร 1 ตัวไปให้เครื่องพิมพ์) เมื่อรันโปรแกรมแล้วรันซ้ำอีกครั้ง จะมีข้อความ 'PRNDRV.COM already' หมายความว่า โปรแกรม PRNDRV.COM ได้รันไปแล้วไม่ต้องรันอีกก็สามารถใช้ได้

การดัดแปลงแก้ไข

สถานะเริ่มแรกของโปรแกรม PRNDRV.COM จะเป็น 'ON' สามารถกำหนดเป็น 'OFF' ได้คือ

- ตัวแปรชื่อ selecode ปกติเป็น 78 (Asc 78 = 'N') เปลี่ยนเป็น 70 (Asc 70 = 'F')
- ตัวแปรชื่อ statusact ปกติเป็น 00 เปลี่ยนเป็น 0FFH

ส่วนสถานะรหัสของเครื่องพิมพ์จะเก็บในตัวแปร codeprn เริ่มแรกจะเป็น สมอ. (80) ถ้าต้องการเป็น เกษตร ให้เปลี่ยนจาก 80 เป็น 75 หรือต้องการให้เป็น Nothing เปลี่ยนเป็น 78

ก็คงทราบกันดีว่ารหัสภาษาไทยนั้นมีหลายรหัส ที่แตกต่างกันออกไป แต่ผู้เขียนมีให้เลือกเพียง 2 รหัส อีกทั้งสองรหัสนี้ก็ยังมีการปรับปรุงแก้ไขใหม่อีก ซึ่งอาจจะไม่ตรงกับค่าที่กำหนดไว้ในโปรแกรม วิธีการแก้ดังนี้ครับ

1. ดูรหัสของเครื่องพิมพ์ตั้งแต่แอสกี้ 128-255 (โดยใช้โปรแกรมที่ 1. เขียนด้วย ปาสคาล หรือโปรแกรมที่ 2. ดังรูปที่ 10 เขียนด้วยเบสิค หรือภาษาอื่น ๆ ก็ได้) ตรวจดูว่าตัวอักษรที่พิมพ์ออกมามีภาษาไทยหรือไม่ ถ้าไม่มีรหัสภาษาไทย ตอนเลือก Printer code ต้องเลือก 'N' othing

2. ถ้าเครื่องพิมพ์ พิมพ์รหัสภาษาไทยออกมาด้วย ดังรูปที่ 11 ตัวที่จะต้องพิจารณาก็คือ ตัวอักษรที่ใช้พิมพ์กรอบเท่านั้น ให้ดูเปรียบเทียบกับค่าที่กำหนดในโปรแกรมชื่อ smocode ซึ่งกำหนดไว้ดังรูปที่ 12 แต่รหัสที่พิมพ์ออกมาได้จากข้อ 2 เป็นดังรูปที่ 13 จะเห็นว่าค่าที่กำหนดไว้ในโปรแกรมและค่าที่เครื่องพิมพ์ พิมพ์ออกมาไม่ตรงกัน เมื่อเป็นเช่นนี้จะต้องเปลี่ยนค่า smocode ใหม่ให้ตรงกับรหัสภาษาไทยที่เครื่องพิมพ์นั้นใช้อยู่ โดยเปลี่ยนดังรูปที่ 14 เป็นอันว่าเสร็จเรียบร้อยครับ แต่ถ้าเครื่องพิมพ์นั้นเป็นรหัสภาษาไทยตัวอื่นอีก ก็ให้เปลี่ยนโดยใช้หลักการเดียวกันนี้

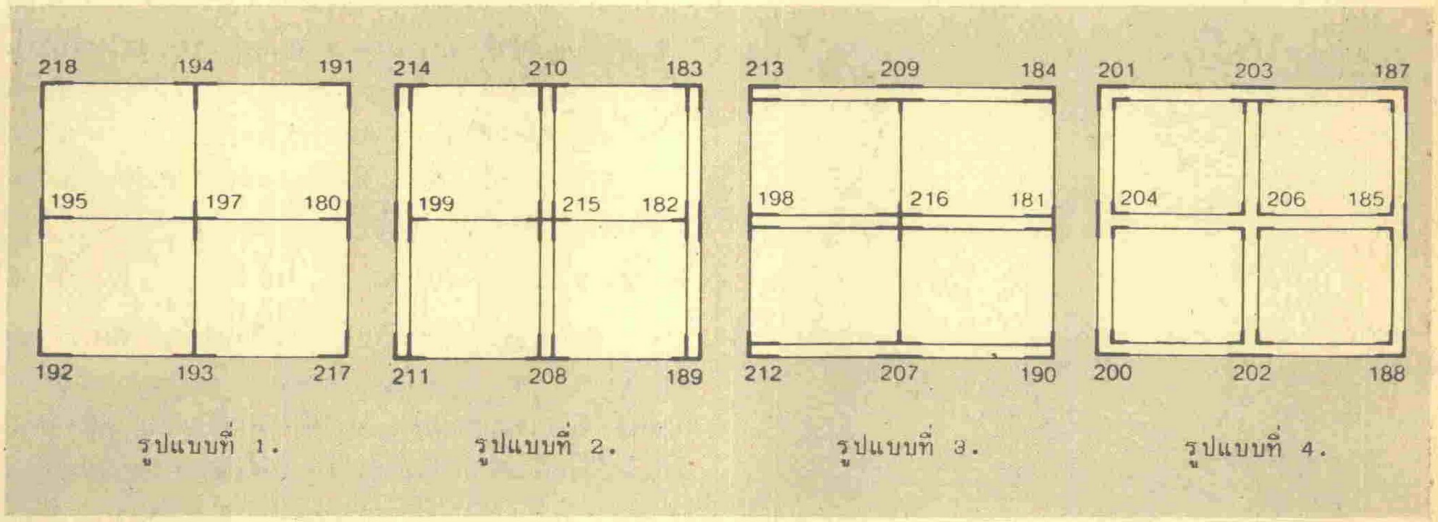
3. Control code ของเครื่องพิมพ์ที่ใช้ในโปรแกรมนี้เป็นของเครื่องพิมพ์ EPSON LX-800 ถ้า Control code ไม่ตรงกับเครื่องพิมพ์ที่ผู้อ่านใช้อยู่ ค่าที่จะต้องเปลี่ยนก็คือ

3.1 bi__direct เป็น Control code ให้เครื่องพิมพ์พิมพ์ไปกลับ

3.2 uni__direct เป็น Control code ให้เครื่องพิมพ์พิมพ์ไปทางเดียว

3.3 left__right เป็น Control code ให้เครื่องพิมพ์พิมพ์จากซ้ายไปขวา

3.4 inch__16 เป็น Control code set line spacing เป็น 1/6"



รูปที่ 9 กรอบทั้ง 4 รูปแบบ

```
PROGRAM  READ_CODEPRN;
VAR
        L : BYTE;
BEGIN
        FOR L := 128 TO 255 DO
          WRITE(LST,L,'(',CHR(L),')    ');
END.
```

โปรแกรมที่ 1. อ่านรหัสของเครื่องพิมพ์ (Ascii 128-255) ใช้ภาษา PASCAL

```
10 FOR L = 128 TO 255
20 LPRINT L;"(";CHR$(L);")    ";
30 NEXT L
```

โปรแกรมที่ 2. อ่านรหัสของเครื่องพิมพ์ (Ascii 128-255) ใช้ภาษา BASIC

รูปที่ 10
โปรแกรมที่อ่านรหัสของเครื่องพิมพ์

```
128(ç) 129(ü) 130(é) 131(â) 132(ä) 133(à) 134(å) 135(ç) 136(ê) 137(ë)
138(è) 139(ï) 140(î) 141(ì) 142(Ä) 143(Å) 144(É) 145(æ) 146(Æ) 147(ô)
148(ö)... 255( )  (เป็นรหัส สมอ. อักรุ่นหนึ่ง)
```

รูปที่ 11

```
smocode  db  152,157,153,156,151,158,154,159,155
                ┌  ┬  ┐  ├  ┼  ┤  └  ┴  ┘
versmo   db 149
horsmo   db 150
```

รูปที่ 12

```
                129,138,130,135,139,136,131,137,132
                ┌  ┬  ┐  ├  ┼  ┤  └  ┴  ┘
versmo   db 133
horsmo   db 134
```

รูปที่ 13

```
smocode  db   129,138,130,135,139,136,131,137,132
                ┌  ┬  ┐  ├  ┼  ┤  └  ┴  ┘
versmo   db 133
horsmo   db 134
```

รูปที่ 14

3.5 inch__672 เป็น Control code set line spacing เป็น 6/72"

แต่ส่วนมากแล้วเครื่องพิมพ์ในตระกูลเดียวกันจะใช้ได้

จอภาพและคีย์บอร์ด

เอดิเตอร์บางตัวไม่สามารถรับ character พิเศษตั้งแต่แอสกี้ 128-255 ได้โดยตรง (ตัวอักษร ที่ใช้พิมพ์กรอบก็อยู่ในข่ายนี้ด้วย เมื่อต้องการแสดงตัวอักษรเหล่านี้ จะใช้ฟังก์ชันหรือ คำสั่งของแต่ละภาษาช่วยจึงจะสามารถแสดงผลออกมาได้ เช่น writeln (chr (179)); จะได้ตัว '|' แต่ถ้าสามารถเขียนได้เป็น writeln ('|'); ทั้งประหยัดทั้งสะดวก โปรแกรม NEWKEY.COM ช่วยทำได้ครับ

หลักการ

หลักการใหญ่ ๆ คือว่า จะต้องเขียนโปรแกรมฝังตัวลงในหน่วยความจำ (resident) คอยตรวจสอบการกดคีย์ ว่าเป็นคีย์ที่ต้องการหรือเปล่า ถ้าไม่ใช่ก็ปล่อยผ่านไปตามปกติ ถ้าใช่ก็เปลี่ยนเป็นตัวอักษรตัวอื่นได้ตามที่ต้องการ เช่น โปรแกรมคอยตรวจจับคีย์ 'A' แล้วเปลี่ยนเป็น '┌' ดังนั้นทุกครั้งที่กด 'A' จะได้เป็นตัว '┌' แทน แต่ถ้าต้องการกดเอาตัว

โปรแกรมย่อย	หน้าที่
CHECK_CAPS	ตรวจสอบและรายงานสถานะของคีย์ CapsLock
CHANGE_KEY	ตรวจสอบคีย์ที่ใช้ในการพิมพ์กรอบ และจะเปลี่ยนเป็นตัวอักษรที่ใช้พิมพ์กรอบด้วย
WAITTING	หยุดรับคีย์จนกว่าจะกด 'ESC'
SAVESCREEN	เก็บจอภาพขณะนั้นไว้ ก่อนที่จะแสดง HELP
RESTSCREEN	คืนจอภาพเดิมให้ หลังออกจาก HELP
SAVECURSOR	เก็บตำแหน่ง cursor เดิมไว้ ก่อนที่จะแสดง HELP
SET_CURSOR	set cursor ให้ไปตำแหน่งที่เก็บไว้ใน row, column
WRITEHELP	พิมพ์ HELP
RESTCURSOR	คืนตำแหน่ง cursor หลังออกจาก HELP

ตารางที่ 4 หน้าที่ต่าง ๆ ของโปรแกรมย่อยของ NEWKEY

'A' ก็จะทำไม่ได้ เพราะจะได้เป็นตัว '┌' แทนตลอด การแก้ปัญหาคือ จะต้องใช้คีย์ Insert, CapsLock, ScollLock หรือคีย์อื่น ๆ ก็ได้ เป็นตัวบอกสถานะว่า จะให้โปรแกรมเปลี่ยนจากตัว 'A' ได้เป็น '┌' หรือไม่ ในโปรแกรม NEWKEY ใช้คีย์ CapsLock เป็นตัวบอกสถานะ ถ้า CapsLock "ติด" จะสามารถรับตัวอักษรที่ใช้พิมพ์กรอบทางคีย์บอร์ดได้ ถ้า "ดับ" การรับคีย์จะเป็นตามปกติทุกอย่าง ไม่เกี่ยวข้องกับโปรแกรม NEWKEY เหตุที่ผู้เขียนไม่ใช้คีย์ ScollLock เป็นตัวบอกสถานะ (น่าใช้มากกว่าคีย์ CapsLock) เนื่องจากถ้าใช้คีย์ ScollLock แล้ว จะทำให้ โปรแกรม NEWKEY ใช้กับ NOTEPAD ของ SIDEKICK ไม่ได้

อธิบายโปรแกรม

- oldvect 16H เก็บ interrupt vector ของ INT 16H ไว้
- capslock เก็บสถานะของคีย์ CapsLock
- position เก็บตำแหน่งของ cursor ก่อนที่จะแสดง HELP
- row, column เก็บตำแหน่ง cursor ที่จะกระโดดไป

หน้าที่ต่าง ๆ ของโปรแกรมย่อยดูในตารางที่ 4

ค่าที่กำหนดไว้เป็นตาราง

- table2 เป็นค่าแอสกี้ที่ใช้พิมพ์กรอบรูปแบบที่ 1,2,3 ดูในรูปที่ 9
- table3 เป็นค่าแอสกี้ที่ใช้พิมพ์กรอบรูปแบบที่ 4 ดูในรูปที่ 9
- โปรแกรมหลัง INSTALL มีหน้าที่ทำให้โปรแกรม NEWKEY ฝังตัวในหน่วยความจำ (reşident)

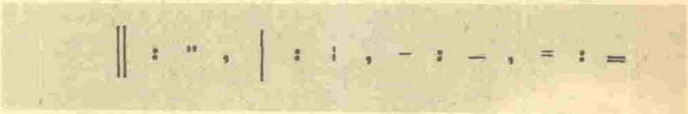
การใช้งาน

เมื่อรันโปรแกรม NEWKEY.COM จะมีข้อความแนะนำโปรแกรมก่อนที่มุมบนด้านซ้ายของจอภาพ การทำงานทุกอย่างก็ยังคงเป็นปกติดี จนกว่าคีย์ CapsLock จะติด จึงจะสามารถใช้คีย์ที่กำหนดไว้พิมพ์กรอบได้ ส่วนคีย์ที่ไม่ได้ใช้ในการพิมพ์กรอบ ยังคงใช้ได้เป็นปกติทุกตัว สามารถเรียก HELP ได้โดยกด Ctrl+F10

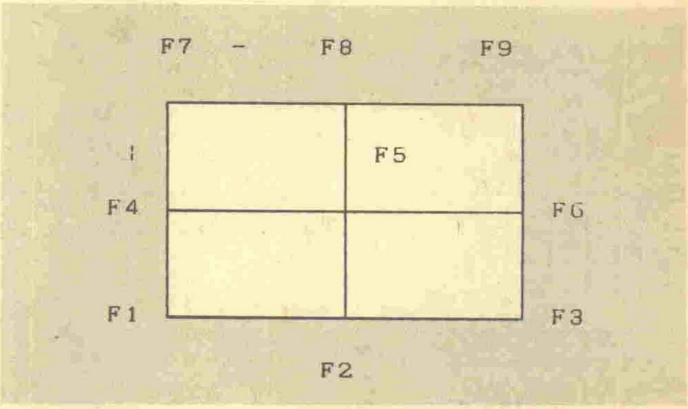
เนื่องจากกรอบจะมี 4 รูปแบบ ดังรูปที่ 9. คีย์ที่ใช้จะต้องมีครบทั้งหมด โดยผู้เขียนจัดไว้ดังรูปที่ 15 คีย์ที่เหลือคือคีย์ที่ใช้ทำมุมดังนี้

กรอบรูปแบบที่ 1 จะใช้ Shift + F1 - F9
กรอบรูปแบบที่ 2 จะใช้ Ctrl + F1 - F9
กรอบรูปแบบที่ 3 จะใช้ Alt + F1 - F9
กรอบรูปแบบที่ 4 จะใช้ Alt + 1 - 9

เช่น เมื่อต้องการพิมพ์กรอบแบบที่ 1 จะต้องกด Shift+F1-F9 ดังรูปที่ 16 จะสังเกตเห็นว่าการวางตำแหน่งคีย์ที่ใช้



รูปที่ 15



รูปที่ 16 การจัดวางตำแหน่งคีย์ที่ใช้พิมพ์กรอบ

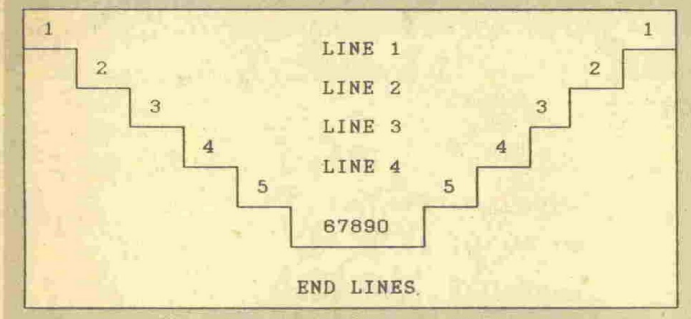
พิมพ์กรอบ จะใช้รูปแบบสี่เหลี่ยมของ Keypads (คีย์ตัวเลขที่อยู่ด้านขวา) ซึ่งจัดวางเป็นรูปสี่เหลี่ยมเหมือนกันพอดี (ใช้ดูเปรียบเทียบเท่านั้น) โดยที่

- 1,3,7,9 ใช้เป็นมุมทั้งสี่ [ล่างซ้าย (└) ล่างขวา (┘), บนซ้าย (┌), บนขวา (┐)] สังเกตุที่ Keypads ว่าจัดวางเหมือนกัน
- 2,4,6,8 ใช้เป็นขอบกลาง [กลางล่าง (┴), กลางซ้าย (├),กลางขวา (┤), กลางบน (┬)]
- 5 ใช้เป็นเครื่องหมายบวกกลาง (┼)
- จะใช้ - แทน ─
- จะใช้ ¦ แทน │

เป็นอันว่าได้กรอบรูปแบบที่ 1 ตามต้องการ ถ้าต้องการกรอบรูปแบบที่ 2 เพียงเปลี่ยนจาก Shift เป็น Ctrl และ '_' ใช้ '-' '═' ใช้ '═' F1-F9 ก็ยังเหมือนเดิม ถ้าต้องการกรอบรูปแบบที่ 3 ก็เปลี่ยนจาก Ctrl เป็น Alt และ '‖' ใช้ '"' ส่วน F1-F9 ก็ยังเหมือนเดิม ถ้าต้องการกรอบรูปแบบที่ 4 จะใช้ Alt เหมือนเดิม แต่จะเปลี่ยนจาก F1-F9 เป็น 1-9

การใช้คีย์คงจะพอเข้าใจ หรือถ้า "งง" กด Ctrl+F10 จะมี HELP (แต่อย่าลืมว่าคีย์ CapsLock ต้องติดก่อน) ให้ดู ข้อเสียของโปรแกรม NEWKEY.COM ก็มีเหมือนกัน คือ เอดิเตอร์บางตัวจะใช้ฟังก์ชันคีย์เดียวกัน หรือตรงกันกับโปรแกรม NEWKEY.COM ใช้ เช่น ในเอดิเตอร์ของ TURBO PASCAL 4.0 คีย์ Alt+F9 จะใช้ในการคอมพายล์ ถ้ากด Alt+F9 ในขณะที่คีย์ CapsLock ดับ จะเป็นการคอมพายล์ตามปกติ แต่ถ้าขณะนั้น CapsLock ติด จะได้ตัวอักษรที่ใช้พิมพ์กรอบ คือ '╡' และมีคีย์อีกตัวหนึ่ง ในเอดิเตอร์ของ TURBO PASCAL 4.0 คือคีย์ Alt+F1 ซึ่งเป็นคีย์ใช้เรียก HELP ของ TURBO PASCAL เอง ซึ่งถือว่ามีความสำคัญมาก จะต้องตอบสนองทุกครั้ง ดังนั้นเมื่อใช้โปรแกรม NEWKEY.COM ร่วมกับ TURBO PASCAL รุ่นนี้ (และรุ่นอื่น ๆ ที่ใช้คีย์เหมือนกัน) จึงจะเอาตัวอักษรพิมพ์กรอบจากคีย์ Alt+F1 คือ '╞' ไม่ได้ ผู้เขียนแก้ปัญหานี้โดยกำหนดให้คีย์ Alt+F10 มีค่าเท่ากับคีย์ Alt+F1 จะใช้ตัวไหนก็ได้ แต่ถ้าใช้กับ TURBO PASCAL 4.0 จะต้องใช้ Alt+F10 จึงจะได้

TEST WINDOW 2



TEST WINDOW 2 <END>

TEST WINDOW 3

NAME	SEX	JOB TITLE	CODE	SALARY	TEL	REMARKS
CAT	M	POLICE	0010	3453	232	—
DOG	M	STUDENT	0020	34534	234	—
RAT	M	PROGRAMMER	0030	67867	456	—
DUCK	W	NURSE	0040	4655	764	—
ANT	W	DOCTOR	0050	7875	434	—
OXEN	M	PILOT	0060	2324	544	—
BIRD	M	DRIVE BUS	0070	24235	644	—
COW	W	POLICE	0080	46757	363	—
FISH	M	PROGRAMMER	0090	23423	235	—
PIG	M	FAMER	0100	23322	643	—

TEST WINDOW 3 <END>

TEST WINDOW 1

1	2	3	4	5	6	7
A H	B I	C J	D K	E L	F M	G N
8	9	A	B	C	D	E

TEST WINDOW 1 <END>

รูปที่ 17 ตัวอย่างการพิมพ์กรอบที่สามารถทำได้

หมายเหตุ

1. ถ้าจะใช้โปรแกรม NEWKEY ร่วมกับ NOTEPAD ของ SIDEKICK จะต้องรันก่อน SIDEKICK และต้องกด Ctrl+QG จะมีคำว่า Graph ขึ้นที่มุมบนด้านขวา NOTEPAD จึงจะยอมรับตัวอักษรพิเศษ โปรแกรม NEWKEY จะเปลี่ยนเฉพาะ รหัสแอสกี้เท่านั้น ส่วน SCAN CODE ค่ายังเหมือนเดิม

2. ทั้งสองโปรแกรมคือ NEWKEY.COM และ PRNDRV.COM ไม่มีความเกี่ยวข้องอะไรกัน ทำงานคนละหน้าที่ NEWKEY จะช่วยให้สามารถพิมพ์กรอบ ที่แสดงบนจอภาพได้ง่าย ส่วน PRNDRV จะช่วยให้พิมพ์กรอบออกทางเครื่องพิมพ์ได้

3. ถ้าจะใช้ NEWKEY.COM ร่วมกับ PRNDRV.COM ควรจะรัน PRNDRV ก่อน NEWKEY

4. รูปที่ 17 ตัวอย่างการพิมพ์กรอบที่โปรแกรม PRNDRV.COM สามารถทำได้

5. โปรแกรม PRNDRV.COM ผู้เขียนใช้กับเครื่องพิมพ์ 'EPSON LX-800'

6. ซอร์สโค้ดและโปรแกรมที่คอมพายล์แล้ว ได้จัดลงในซอฟต์เมล์ฉบับนี้ ซึ่งมีดังนี้คือ NEWKEY.COM NEWKEY.COM, PRNDRV. ASM, PRNDRV.COM พร้อมกันนั้นก็มีโปรแกรม TEST1-TEST5 ซึ่งเป็นโปรแกรมตัวอย่างในการทดสอบโปรแกรม PRNDRV.COM ด้วย